

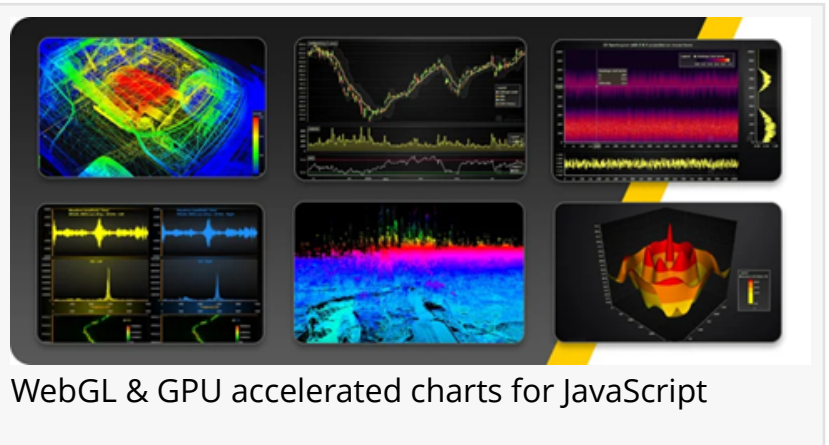
# Understanding WebGL in JavaScript Charts - DELETED

KUOPIO, FINLAND, May 16, 2022

/EINPresswire.com/ -- When JavaScript charts are required to render high-performance end-user data visualization applications, they are obligated to use state-of-the-art technology that can support fast rendering and refresh rates.

What is “state-of-the-art” tech?

WebGL has been introduced as the ultimate solution for web charts. See, WebGL accelerates the rendering process using the client’s GPU. WebGL is easy to use and even faster than traditional HTML5 charts. WebGL can handle dynamic colour lookup by any data point property and 3D projections, oppositely to canvas charts that cannot do this.



WebGL & GPU accelerated charts for JavaScript

With WebGL, it is easier to specify with code what the program or application should create. This can create results that simply cannot be achieved when using simpler SVG or canvas.

High-performance web charts fall under the category of productivity software, yet they are more sophisticated. Web charts use [data visualization APIs](#) to generate plots and graphs in seconds. Here’s an example: WebGL-rendered and GPU-accelerated [Static Surface JavaScript charts](#) can render 4 million data points in as little as 152 MS (milliseconds).

How does that compare to non-hardware accelerated charts? Naturally, non-hardware accelerated charts are slower and limited in performance. Compared to a WebGL and GPU-accelerated JS Static Surface chart, a non-hardware accelerated one would render the same 4 million data points in 22,659 MS (milliseconds). The difference is huge.

## JavaScript Files

IIFE JavaScript files are featured in the most commonly used charts. These files are convenient because once you attach the file to an HTML page, it can be accessed from that moment on. Another benefit is that JavaScript files often do not require any installation or downloading and can be downloaded from the host website in your HTML file.

One way to use web charts is the descriptive invocation approach when the user creates a JSON configuration for the chart. Another approach is a functional app, where the user types code in a terminal to access real-time data. For instance, when creating a [vibration analysis application](#) with LightningChart JS and TypeScript in Visual Studio, a developer would need to simply install the NPM command, run a new terminal in VS and run the NPM START command to see the vibration analysis application running on localhost 8080.

Pasi Tuomainen

Arction Ltd.

+358 453150905

[email us here](#)

Visit us on social media:

[Facebook](#)

[Twitter](#)

[LinkedIn](#)

---

This press release can be viewed online at: <https://www.einpresswire.com/article/572630848>

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.

© 1995-2022 Newsmatics Inc. All Right Reserved.